Development of SGABU Platform for Multiscale Modeling

Nikolić, Jovana; Atanasijević, Aleksandar; Živić, Andreja; Šušteršič, Tijana; Ivanović, Miloš; and Filipović, Nenad

Abstract: ***SGABU platform was created as a computational platform for multiscale modelling in biomedical engineering. This is one of the few proposed integrated platforms that include different areas of bioengineering. The platform includes already developed solutions, various datasets and models related to cancer, cardiovascular, bone disorders, and tissue engineering. The biggest obstacle in designing a platform of this type is the use of different tools for each of the layers of architecture for models which are created using different technologies and their integration and visualization within a platform. This study describes the technologies that were used for building the platform and methods for data and models visualization. The goal was to build the most flexible system capable of executing tools of various nature and connecting them into a platform.***

Index Terms: ***bioengineering, bone modeling, cancer modeling, cardiovascular disease modeling, platform, tissue engineering***

1. Introduction

BIOENGINEERING and biomedical research is becoming increasingly dependent on platforms that have the characteristics of rapid availability and scalability. Progress in bio-engineering and biomedical research is driven by insight gained through the large amount of biomedical data that has been generated and collected at an unprecedented speed and scale.

There are some examples of successful platforms for biomedical research. For example, PANBioRA, a modular platform that standardizes the evaluation of biomaterials and opens the venue for pre-implantation and personalized diagnostics for biomaterial-based applications [1]. The SILICOFCM platform is a computational platform innovative in silico clinical trials solution for the design and functional optimization of whole heart performance and monitoring effectiveness of pharmacological treatment, integrated into the cloud [2]. The Bioengineering and Technology (BET) platform is primarily focused on the needs of the cancer research community and enables innovation, support research and federates the wider interdisciplinary community active in translational bioengineering [3].

This platform is developed not only for research purposes but to integrate teaching materials for various fields. SGABU is a web-based platform that is available for users to access services and tools for biomedical research. In this paper, we present an overview of platform architecture, its features and the examples provided to illustrate the diverse data content, infrastructure, services, tools and methods for increasing access and use of biomedical research data and models. SGABU platform requires interconnecting several various tools into repeatable data-intensive pipelines of workflows. The integration of the platform is performed according to the standardized procedures, following the integration plan. The SGABU architecture will be presented in the following parts of this paper, along with open-source technologies that were used to support all the functionalities and several services.

2. Deployment of the Platform

Deployment of the platform and its architecture are presented in this section.

2.1 Architecture

The SGABU platform architecture is presented in the following Figure 1 with the modules and

This research is funded by Serbian Ministry of Education, Science and Technological Development [451-03-9/2021-14/200107 (Faculty of Engineering, University of Kragujevac)]. This research is also supported by the project that has received funding from the European Union's Horizon 2020 research and innovation programmes under grant agreement No 952603 (SGABU project). This article reflects only the author's view. The Commission is not responsible for any use that may be made of the information it contains.

J.Nikolić is with Bioengineering Research and Development Center (BioIRC), Kragujevac, Serbia (e-mail: joxikg@gmail.com).

A. Atanasijević is with University of Kragujevac Computing Center (e-mail: aatanasijevic@kg.ac.rs).

T. Šušteršič and N. Filipović is with the Faculty of Engineering, University of Kragujevac and Bioengineering Research and Development Center (BioIRC), Kragujevac, Serbia (e-mail: tijanas@kg.ac.rs, fica@kg.ac.rs).

A.Živić is with Faculty of Science, University of Kragujevac (e-mail: andrejazivic2@gmail.com).

M.Ivanović is with Faculty of Science, University of Kragujevac and Bioengineering Research and Development Center (BioIRC), Kragujevac, Serbia (e-mail: mivanovic@kg.ac.rs).

their corresponding engines and tools. The SGABU framework can be defined as a hierarchical multilayer schema. The framework comprises five layers. At the bottom lies the hardware layer where CPUs, RAM, and VMs are specified. Above the hardware layer lies the security layer with additional mechanisms for user access management, encrypted communication and user authentication within the platform. The next layer corresponds to the workflow layer which is the core of the services. The workflow layer includes the following engines: (i) the workflow engine, (ii) the Docker container engine, (iii) the data quality control engine and (iv) the visual analytics engine. Each one of these engines interacts directly with the SGABU tools and modules that lie on the back-end layer. The last layer is the front-end layer which represents the user interface.

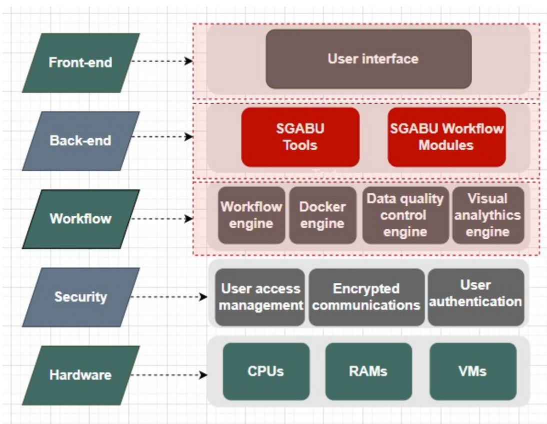

Figure 1: SGABU architecture

Common services are used by both Development and Testing. *DBServer* provides a standard relational MySQL (MariaDB) database. The most resource-demanding component is *FunctionalEngineServer*, responsible for the execution of the CWL [4] (*Common Workflow Language*) compatible scientific workflows. The virtual machines that build the Development deployment are identical to their counterparts in Testing (Figure 2). The services that belong to the Common Services group are already well-established, adopted, and thoroughly tested modules. They do not require a separate development branch due to a slower pace of development compared to the core SGABU modules. The platform communication between different engines and tools is presented in Figure 3.

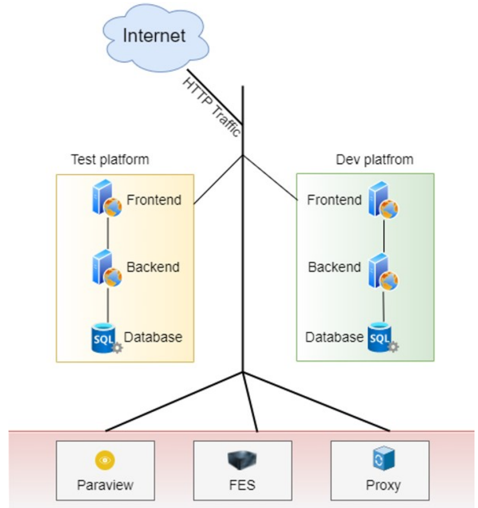

Figure 2: Logical architecture of SGABU platform

The guest proxy server receives a request from the frontend and forwards it to the FES module, which returns a file (or multiple files) depending on the type of request. All HTTP traffic is routed by a central NGINX reverse proxy and is SSL encrypted. Paraview Glance [5] sends a request to the proxy server, which it forwards to FES and then FES returns a list of files that are displayed by Paraview Glance. The frontend sends a request to the proxy server whenever it needs to display an image. The backend creates and controls the work for FES. It receives results and a list of files that are used by FES for processing.

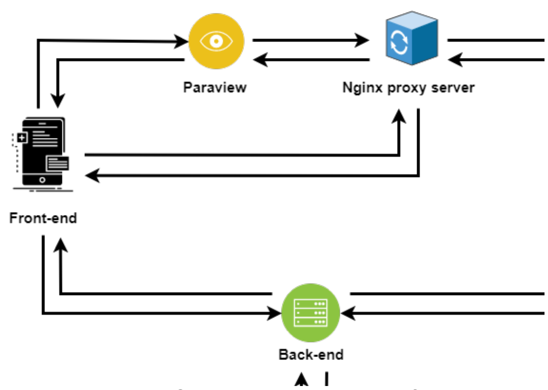

Figure 3: Communication on the platform

2.2 Hardware

The hypervisor cluster consists of 96 CPU cores, 376 GB RAM and 1.89 TB SSD, interconnected with a 1 Gbps internal network, separate management network and the uplink. It runs KVM (Kernel-based Virtual Machine) under Proxmox VE 7 environment.

2.3 Frontend

As one of the most popular software development tools available, Angular [6] is used for building single-page client applications using HTML and TypeScript. This framework implements core and optional functionality as a set of TypeScript libraries that are imported into applications.

The core value proposition of Angular is to make it possible to build applications that work for nearly any platform - whether mobile, web, or desktop. SGABU frontend is developed responsively. It means that various devices (desktop computers, laptops, tablets, smartphones) can correctly render offered web pages. On small screens panels, forms and modals are minimized, while on big screens they are maximized. Thus, responsive web platforms are automatically modified for various web browsers and screens.

Some models or datasets in the SGABU platform produce outputs intended for visualization with the goal of better understanding. Plotly.js [7] is implemented for interactive data visualization and it supports various graphs like line charts, bar charts, scatter plots, area plots, histograms, etc. Plotly uses javascript to make interactive plots where it is possible to zoom in on the graph or add additional information like data on hover. It allows endless customization of graphs that makes plots more meaningful and understandable for users.

2.4 Backend

Laravel [8] is a free and open-source PHP framework that provides a set of tools and resources to build modern PHP applications. It is used as a backend framework for the SGABU platform. It follows a model-view-controller design pattern, which generally makes it a lot easier to start creating and after that, to maintain the functionality of the platform. Additionally, on the SGABU platform, important built-in features like authentication, sessions, routing, migration system are provided.

2.5 FES API

The FES (*Functional Engine Server*) [10] API is a REST API developed in Python whose main purpose is to run and manipulate various workflows. FES-API implements the required executor for running workflows. The FES in the background employs Common Workflow Language (CWL) which provides portability of the workflows. It is possible to manage the entire lifecycle of a single workflow using the functional server interface, including the creation, handling of inputs and outputs. FES is in charge of providing input files to the workflow as well as storage space where the workflow stores its outputs. Additional communication between the workflow framework and the system file takes place during execution. Each workflow must have a repository (also provided by FES), where it will store intermediate data that is later passed through the workflows. Intermediate data is not permanent but exists only during the execution of the workflow and will be removed after its completion.

The FES server is in charge of the entire workflow cycle, as well as execution monitoring. The workflow lifecycle consists of the following phases:

1. Create
2. Start
3. Execute
4. Download output
5. Remove workflow

FES classifies workflows in several different states or statuses they have:

- **`not_yet_executed`**
- **`terminated`**
- **`running`**
- **`finished_ok`**
- **`finished_error`**

The significant advantage of this system is an asynchronous execution of workflows as background processes, which allows multiple workflows to run simultaneously. The first phase of the workflow cycle is the creation of the workflow. While creating a workflow, the service must be provided with a "familiar" CWL template and input parameters. The space where the input data will be stored, as well as the space where the workflow will store the results, is provided. If the status of the workflow execution is successful and all the necessary parameters have been passed and validated, the creation status is returned. Starting a workflow is a simple transition from **`not_yet_executed`** to **`running`** state. Another important feature is downloading the entire workflow results. Removing a workflow involves deleting the workflow from the system, except the workflow template.

2.6 Paraview

SGABU platform integrates the Paraview Glance [5] module, as a 3D visualizer that is utilized in many simulation platforms. The Paraview Glance module accepts the standard file types as input files and allows the end-user to have a quick “glance” of the simulation results.

The 3D visualization module is an integrated version of the ParaViewWeb [9] (the JavaScript library) which is a Web framework to build applications with interactive scientific visualization inside the Web browser. A number of models in the SGABU platform produce outputs intended for 3D visualization. However, different tools create different formats that create the problem of interoperability between the tools. One of the common formats for storing 3D simulation results is VTK files which can be handled by any post-processing software and the Paraview Glance 3D [5] visualization framework adapted for the SGABU platform needs.

2.7 UML Diagrams

The purpose of the following UML diagrams is to visually present the platform along with its main actors, roles, actions, and artefacts, in order to document the needed information about the system. The sequence diagram shows the interactions between objects in the sequential order of these interactions. The sequence diagram for a user's login is presented in Figure 4. The credentials are passed through the login form to the API which starts the validation of user credentials. The API communicates with the Database for user validation and if the credentials are valid, the API retrieves the access token.

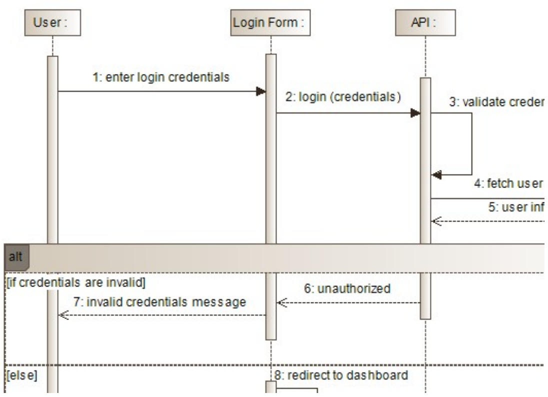

Figure 4: Sequence diagram for user's login

The sequence diagram presented in Figure 5 refers to the registration of a new user. On the register page, a user enters the new user information, which is taken by the user registration page and is sent to the API. If the validation is successful, the API creates a new user in the database.

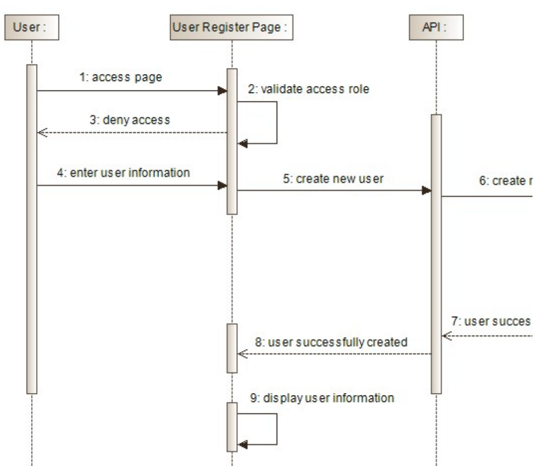

Figure 5: Sequence diagram for the registration of a new user

The sequence diagram presented in Figure 6 refers to the workflow creation. When a user selects a specific workflow type from the dashboard, the workflow creation sequence starts. After selecting the input parameters which are validated on the workflow page, the workflow page sends the data to the FES server. The FES server returns a success message regarding the start of the workflow. The workflow page checks the workflow status from the FES, and after execution of the workflow, the FES sends the data to the workflow page.

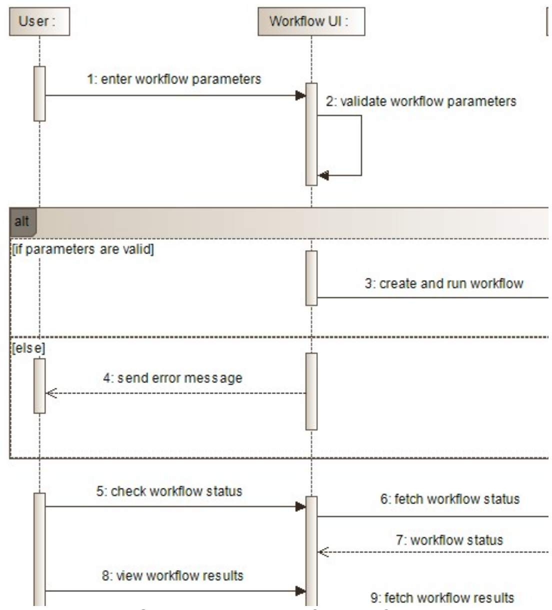

Figure 6: Sequence diagram for workflow creation

3. DEMONSTRATION OF THE PLATFORM

A SGABU login page requires the user's access credentials for the user to have access to the platform. The user is also able to sign in with his/her Google account. The final user login form is presented in Figure 7.

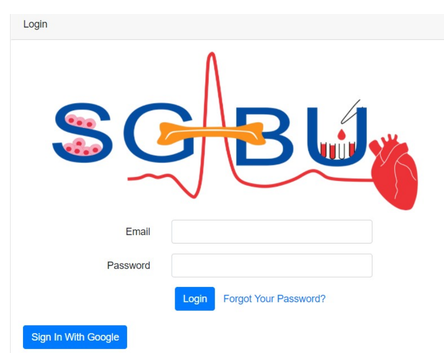

Figure 7: SGABU login page

A new user can register to the platform through the registration page and after the administrator's authorization, provide his personal information such as email, first name, last name, and password (Figure 8).

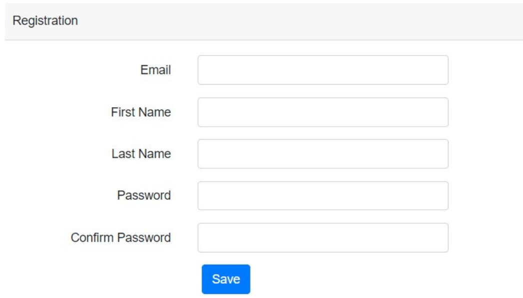

Figure 8: SGABU registration page

When the user registration is accepted by the administrator, the user can check his profile and personal information. Only the administrator of the platform has access to the user list and he is responsible for activating or deactivating the account of the user (Figure 9).

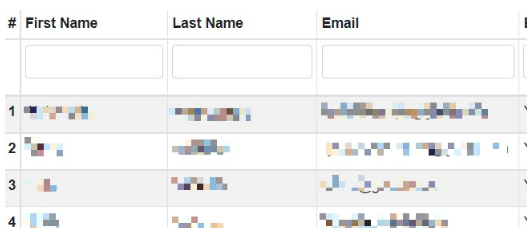

Figure 9. User list

After the login process, the end-user has gained access to the main page of the platform. The main user interface structure consists of four submodules and integrates sections of bone modeling, cancer modeling, cardiovascular disease modeling, and tissue engineering. As an example, two models have been presented in sections of bone modeling and cardiovascular disease modeling. Access to these modules on the SGABU platform is provided through the main dashboard. Figure 10 shows the dashboard and presents all the available modules for quick access to the user. Each entry is annotated with the name of the model and model's logo and direct links to the respective subpage so that the user can directly be forwarded to the desired submodule.

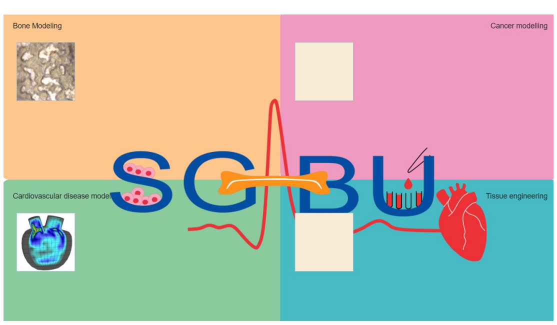

Figure 10: Main dashboard

A parametric heart model of the left ventricle is used to simulate the cardiac cycle with patient-specific dimensions, which are provided by the user. User Interface for ParametricHeart is divided into 2 sections – Workflows and Add new workflow section. In the Workflows section of the window, users can see the names and statuses of the workflows (Figure 11).

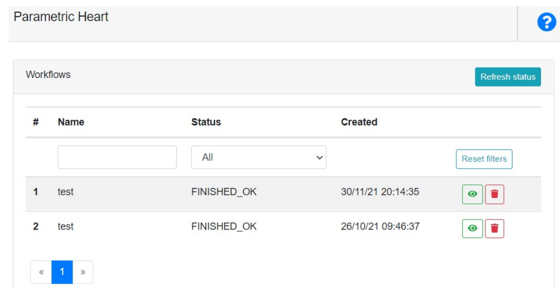

Figure 11: List of workflows section on the window

In the Add new workflow section, users can create new workflows for this submodule. The exception handling is integrated into the user interface (UI) of the SGABU platform (empty forms, non-numerical forms, out of range values). Once everything is correctly filled, the workflow can be started. Results are displayed in the form of tables, data, plots, video and Paraview. The results can also be downloaded in the form of .csv files.

Clicking on the Paraview tab opens a new browser tab with results. In Figure 12, results on velocity distribution of the left ventricle are shown.

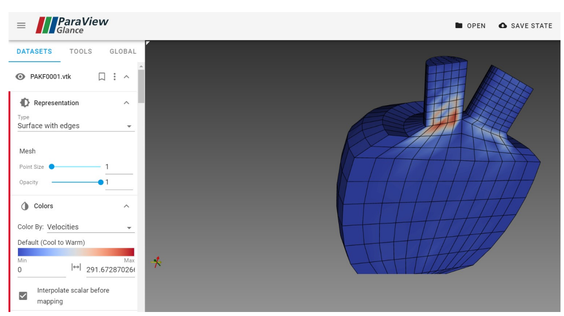

Figure 12: Paraview tab of the results section

4. Conclusion

In this paper, a platform for adaptive knowledge discovery of multiscale models and different datasets is provided. The major advantage is using new, modern, and different technologies for different layers of architecture while building a platform. These technologies are completely independent of the operating system, which allows easy transfer of the platform from the virtual machine to the public cloud if the need arises. There are many advantages for users who are using SGABU. Users can directly use the platform without any new software installations on their local devices because SGABU platform runs within a web browser. We anticipate that new insights in area research will require a platform to support the integration of multiple data types for research projects and for practical application.

REFERENCES

[1] PANBioRa, Accessed: 15.12.2021., Available: https://www.panbiora.eu/

[2] SILICOFCM, Accessed: 15.12.2021., Available: https://silicofcm.eu/

[3] Bioengineering and Technology platform – BET, Accessed: 15.12.2021., Available: https://www.epfl.ch/research/facilities/ptbet/

[4] Common Workflow Languarge, Accessed: 15.12.2021., Available: https://www.commonwl.org/

[5] Paraview-Glance, Accessed: 20.12.2021., Available: https://kitware.github.io/paraview-glance/index.html

[6] Angular, Accessed: 18.12.2021., Available: https://angular.io/

[7] Plotly, Accessed: 18.12.2021., Available: https://plotly.com/

[8] Laravel, Accessed: 18.12.2021., Available: https://laravel.com/

[9] ParaViewWeb, Accessed: 19.12.2021., Available: https://www.paraview.org/web/

[10] Ivanovic, M., Zivic, A., Tachos, N., Gois, G., Filipovic, N., Fotiadis, D., "In-silico Research Platform in the Cloud - Performance and Scalability Analysis", pp. 1-6, 2021.